Université de Limoges | FACULTÉ DES SCIENCES ET TECHNIQUES

Master 1ère année

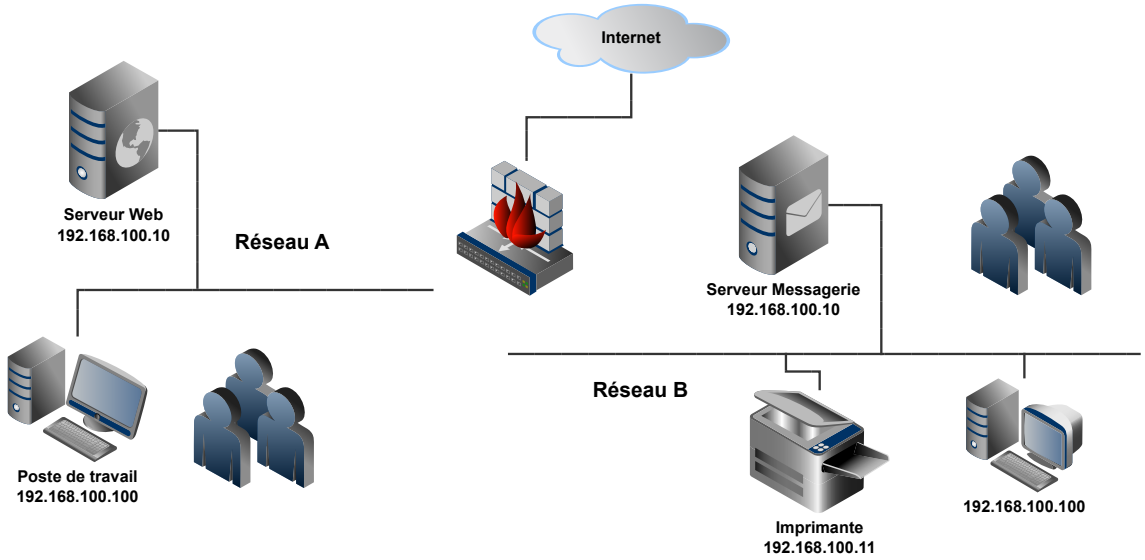
UE Réseaux Avancés I

Corrections — décembre 2012

Durée : 2h — Documents autorisés

NetFilter & Gestion de congestion (12 points)

1 – 5pts Le responsable d'une petite entreprise vous appelle pour vous confier la tâche de « fusionner » le réseau de son entreprise avec celui d'une autre entreprise qu'il vient de racheter.



Les deux réseaux sont organisés de la même manière :

- le réseau A : en `192.168.100.0/24` :
  - un serveur Web est accessible à l'adresse `192.168.100.10` ;
  - les postes clients possèdent des adresses à partir de l'adresse `192.168.100.100` ;
- le réseau B : également en `192.168.100.0/24` :
  - une imprimante, qui permet d'imprimer par TCP sur le port 9100, est accessible à l'adresse `192.168.100.11` ;
  - un serveur de messagerie interne, accessible en POP et en SMTP, est accessible à l'adresse `192.168.100.10` ;
  - les postes clients possèdent des adresses à partir de l'adresse `192.168.100.100` ;
- les deux réseaux ne disposent d'aucun accès vers Internet.

Un routeur sous Linux a été acheté afin de servir de routeur et de firewall entre les deux réseaux, ainsi que de passerelle vers Internet.

Le responsable vous expose les contraintes :

- il ne faut pas modifier la configuration réseau des matériels connectés dans les réseaux A et B ;
- les utilisateurs du réseau A doivent pouvoir accéder au serveur de messagerie et à l'imprimante du réseau B ;
- les utilisateurs du réseau B doivent pouvoir accéder au serveur Web du réseau A ;

a. Est-il possible, grâce aux possibilités de NetFilter, de proposer une solution ?
Quels types d'opérations vous allez mettre en place, et comment les utilisateurs accéderont aux services proposés dans l'autre réseau que le leur ?
Tout d'abord, on remarque que les deux réseaux resteront indépendants après la mise en place du routeur, il n'y aura donc pas de « fusion » entre les deux réseaux, ce qui nécessite la présence du routeur pour les faire communiquer entre eux.

La solution utilisant NetFilter est la suivante :

- ◇ *pour les machines du réseau A, le routeur sera considéré « comme » le serveur hébergeant le serveur de messagerie et la file d'impression de l'imprimante. Ce qui veut dire que les machines du réseau A se connecteront sur le routeur sur un port choisi et le routeur relaiera la communication vers le serveur offrant réellement le service : on réalisera du **DNAT**.*
- ◇ *pour les machines du réseau B, le routeur sera considéré « comme » le serveur hébergeant le serveur Web. De même, on réalisera du **DNAT** pour atteindre le serveur hébergeant réellement le service.*
- ◇ *en prenant l'exemple d'une machine du réseau B, voulant accéder au serveur Web du réseau A, on remarque que si la destination sera réécrite de manière correcte par le **DNAT** pour un datagramme IP de $Machine_B \Rightarrow Serveur_{Web}$, il sera nécessaire de transmettre les datagrammes IP **en retour** de $Serveur_{Web} \Rightarrow Machine_B$.*
  *Cette réponse devra être prise en charge par le routeur lui-même, en **reprenant à son compte**, la communication, c-à-d en réalisant du **SNAT**. On choisira avec soin l'adresse source de ce SNAT en la faisant correspondre à l'adresse de l'interface du routeur connectée au réseau de **destination** initiale de la connexion.*
- ◇ *une configuration du SNAT et du DNAT est nécessaire pour chaque service à relayer au travers du routeur.*

b. Donnez la configuration complète du firewall, après avoir choisi une adresse de connexion pour le routeur/firewall dans le réseau A et dans le réseau B.

*Il est **impossible de conserver** la configuration initiale des postes à cause du **routage** :*

- ◇ *sachant que le routeur sera connecté aux deux réseaux simultanément, que le routage se fait suivant le préfixe le plus long, partagé entre une adresse destination et une entrée de la table de routage, il sera nécessaire de configurer le routeur :*
  - ⋆ *avec deux adresses IP différentes pour sa connexion au réseau A et pour celle au réseau B ;*
  - ⋆ *en lui ajoutant une entrée dans sa table de routage pour **chaque serveur individuellement**, et en tenant compte de l'interface d'accès au réseau qui le contient.*
    On remarque qu'il faut modifier la configuration du serveur Web et de Messagerie qui possède la même adresse IP, sous peine de rendre impossible le routage sur le routeur !
    On choisira alors l'adresse IP `192.168.100.9` *pour le serveur Web et on conservera l'adresse IP* `192.168.100.10` *pour le serveur de messagerie.*
  - ⋆ *en lui ajoutant également une entrée dans sa table de routage pour **chaque poste de travail individuellement** pour la méme raison (nécessité d'indiquer pour une adresse donnée l'interface d'accès).*
    Il faut que les plages d'adresses des postes de travail du réseau A et du réseau B :
    - ▷ ***soient disjointes, par exemple*** `192.168.100.100-199` ***pour le réseau B et*** `192.168.100.200-299` ***pour le réseau A ;***
    - ▷ ***soient entrées sous forme d'une liste d'adresses individuelles (une pour chaque machine connectée), dans la table de routage du routeur.***

On configure le routeur de la façon suivante (on imagine que les interfaces reçoivent les noms `eth0` *et* `eth1`*) :*

- ◇ *interface* `eth0` *connectée au Réseau B, configurée en* `192.168.100.254` *;*
- ◇ *interface* `eth1` *connectée au Réseau A, configurée en* `192.168.100.253` *;*

On ne décrira que la configuration d'accès du réseau B vers le serveur Web du réseau A, les autres services étant configurés de la même façon.

- ◇ *on active le routage sur le routeur :*

```
# sysctl net.ipv4.ip_forward=1
```

- ◇ *on ajoute l'entrée suivante dans la table de routage du routeur :*

```
# ip route add 192.168.100.9 dev eth1
# ip route add 192.168.100.100 dev eth0
```

- ◇ *le poste* `192.168.100.100` *du réseau B voulant accéder au serveur web :*

```
$ socat - tcp:192.168.100.254:80
```

Il désigne le routeur comme étant la cible de sa connexion.

◇ *On ajoute la règle de DNAT suivante dans le routeur :*

```
# iptables -t nat -A PREROUTING -d 192.168.100.254 -p tcp --dport 80 -j DNAT
  --to-destination 192.168.100.9:80
```

On remarque que la table de routage contient une route de préfixe complet, `/32`*, vers l'interface* `eth1`*, ce qui permet au routeur de choisir la bonne interface de sortie.*

◇ *On ajoute la règle de SNAT suivante dans le routeur :*

```
# iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to 192.168.100.253
```

La source du datagramme désigne maintenant le routeur comme origine, ce qui permet au serveur du réseau A de répondre vers le routeur.

c. Le responsable désire maintenant donner l'accès à Internet pour tous les utilisateurs des deux réseaux. Donnez les règles de firewall à ajouter.
On considère que le routeur dispose d'une troisième interface eth2 et que les postes des réseaux A et B possèdent l'interface du routeur associé comme passerelle de sortie :

```
# iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

d. Le responsable voudrait maintenant donner l'accès à Internet **uniquement** aux utilisateurs du réseau A. Est-ce possible ? et si oui, donnez la configuration du firewall le permettant.
Il faut enlever la règle de la question précédente et ajouter une règle pour chaque machine de la plage `192.168.100.200-299`*, comme par exemple :*

```
# iptables -t nat -A POSTROUTING -s 192.168.100.200 -o eth2 -j MASQUERADE
```

2 – 4pts Soit la configuration suivante de NetFilter sur la machine A `192.168.127.186` (seule la table « `filter` » est utilisée) :

```
Chain INPUT (policy DROP 198 packets, 10591 bytes)
 pkts bytes target     prot opt in     out     source         destination
    6   321 ACCEPT     tcp  --  *      *       192.168.127.1  0.0.0.0/0     tcp dpt:50000 limit: avg 1/min burst 6

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source         destination

Chain OUTPUT (policy ACCEPT 175 packets, 8030 bytes)
 pkts bytes target     prot opt in     out     source         destination
```

a. Décrivez l'effet de ces règles.
Tout d'abord, il y a une « policy » DROP sur la chaîne INPUT de la machine `192.168.127.186`*, c-à-d que tout datagramme à destination explicite de cette machine est détruit.*
Ensuite, la première règle de la chaîne INPUT, autorise des datagrammes en provenance de `192.168.127.1` *pour le protocole TCP à destination du port* 50000 *uniquement pour au plus 6 datagrammes d'un coup, si le « bucket » associé à la règle a eu le temps de se remplir de 6 jetons, ou pour seulement un datagramme par minute.*

b. On lance la commande suivante sur la machine A hébergeant le firewall :

```
$ socat stdio tcp-listen:50000
```

Est-ce que si l'on lance la commande suivante sur la machine B `192.168.127.1` :

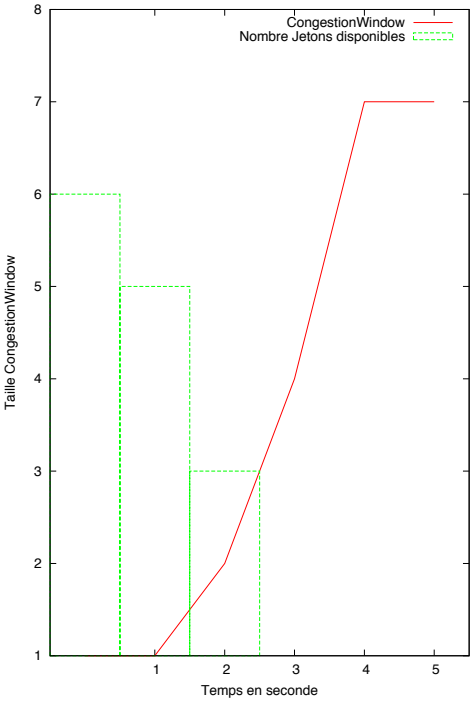
```
$ socat stdio tcp:192.168.127.186:50000
```

la connexion s'établiera ?
Oui, la connexion pourra s'établir si le « bucket » de la première règle de la chaîne INPUT est plein, car le « handshake » TCP ne consomme que 2 jetons.
D'autre part, les datagrammes en réponse passe par la chaîne OUTPUT et sont, suivant la « policy » associée, autorisés.

c. La machine B veut envoyer $18000ko$ vers la machine A, après une connexion réussie.
Décrivez comment va évoluer dans les 5 premières secondes de communication la fenêtre de congestion en considérant que :
- la machine B utilise l'algorithme Tahoe ;
- le MSS est de $1500ko$;

- la taille de la fenêtre de réception de A est de taille $20ko$;
- le RTT est de $1sec$;
- le RTO sur B est de $3sec$.

Pour transmettre $18000Ko$ *avec un MSS de* $1500Ko$*, il faut 12 segments TCP.*
La présence de la règle de firewall va autoriser uniquement 6 paquets à passer durant les 5 secondes d'observation, ce qui donne l'évolution suivante pour la fenêtre de congestion $\Longrightarrow$



*La fenêtre de congestion, après 5 secondes, atteint la taille de 7*MSS, ce qui est du à la destruction du* $7^{ème}$ *paquet du fait de l'absence de jeton disponible.*

- ◇ temps 1s : envoi du SYN, $T_{CongWin} = 1$;
- ◇ temps 2s : réception du SYN/ACK, $T_{CongWin} = 2$;
- ◇ temps 2s : envoi du ACK + segment, $T_{CongWin} = 2$;
- ◇ temps 3s : réception acquittements de 2 segments, $T_{CongWin} = 4$;
- ◇ temps 3s : envoi de 4 segments, $T_{CongWin} = 4$;
- ◇ temps 3s : **le firewall détruit le dernier segment n°6** ;
- ◇ temps 4s : réception d'acquittements de 3 segments, $T_{CongWin} = 7$;
- ◇ temps 4s : envoi de 7 segments. . . mais plus rien ne passe !

Le RTO étant de 3s et la perte étant survenue à la 3e seconde, il n'a pas encore le temps de se déclencher.

3 – Étude et correction de fichier de configuration de firewall

3pts a. Soit le fichier de configuration de firewall suivant :

```
1 sudo iptables -t filter -F
2 sudo iptables -t filter -P FORWARD DROP
3 sudo iptables -t filter -A FORWARD -p tcp --dport 80 --syn -j ACCEPT
```

Est-ce qu'il est correct pour autoriser les connexions vers le service HTTP ? Si non, pouvez-vous le corriger ?

Il manque la règle autorisant les datagrammes associés à la connexion :

```
1 sudo iptables -t filter -A FORWARD -m state --state ESTABLISHED, RELATED -j ACCEPT
```

b. Soit le fichier de configuration suivant :

```
1 sudo tc qdisc add dev eth0 root handle 1: htb default 10
2 sudo tc class add dev eth0 parent 1: classid 1:10 htb rate 30mbps
3 sudo tc class add dev eth0 parent 1: classid 1:20 htb rate 20mbps
4 sudo tc class add dev eth0 parent 1: classid 1:30 htb rate 50mbps
5 sudo tc filter add dev eth0 protocol ip parent 1: handle 1 fw classid 1:10
6 sudo tc filter add dev eth0 protocol ip parent 1: handle 2 fw classid 1:20
7 sudo tc filter add dev eth0 protocol ip parent 1: handle 3 fw classid 1:30
```

Donnez la règle de firewall permettant de limiter le trafic à 20mbps, en provenance de la machine `193.50.87.18` et à destination d'Internet pour le protocole TCP à destination du port 5900.

Il faut la règle de marquage des datagrammes et celle de filtrage par traffic control :

```
1 sudo iptables -t mangle -A PREROUTING -s 193.50.87.18 -p tcp --dport 5900 -j MARK
--set-mark 1
2 tc filter add dev eth0 protocol ip parent 1:0 prio 1 handle 1 fw flowid 1:20
```

Analyse de trame (3 points)

4 – 3pts Soit la trame suivante :

```
0000   FF FF FF FF FF FF 00 D0  F1 10 12 13 81 00 00 01   ................
0010   08 06 00 01 08 00 06 04  00 01 00 D0 F1 10 12 13   ................
0020   C0 A8 01 79 00 00 00 00  00 00 A4 51 01 04         ...y.......Q..
```

a. Analysez cette trame : que contient-elle ?
Elle contient une requête ARP :

```
<Ether  dst=ff:ff:ff:ff:ff:ff src=00:d0:f1:10:12:13 type=0x8100 |
<Dot1Q  prio=0L id=0L vlan=1L type=0x806 |
<ARP  hwtype=0x1 ptype=0x800 hwlen=6 plen=4 op=who-has hwsrc=00:d0:f1:10:12:13 ↵
  psrc=192.168.1.121 hwdst=00:00:00:00:00:00 pdst=164.81.1.4 |>
```

b. Où cette trame circule-t-elle, entre quels matériels ?
Cette trame est dans une encapsulation VLAN, elle circule donc dans un « trunk » reliant deux switch gérant des VLANs.

c. Est-ce que cette trame est « normale » ou a-t-elle été « forgée » ?
Elle a été forgée : les adresses IPs de source et destination de la requête ARP ne sont pas dans le même réseau local... Le « hacker » est un amateur !

Programmation Python (5 points)

5 – 5pts Le responsable du système d'information d'une entreprise vous demande de programmer un outil permettant à des connexions TCP issues de postes de travail de traverser automatiquement un firewall :

- ▷ le réseau où se trouve les postes de travail est le `10.10.0.0/16` ;
- ▷ le routeur/firewall possède l'adresse IP `10.10.10.10` dans ce réseau ;
- ▷ un serveur logiciel TCP attend sur le routeur/firewall sur le port 6800 et exécute le protocole suivant :
  - ◇ il attend la connexion d'un poste de travail ;
  - ◇ lorsqu'une connexion se réalise, le serveur dispose d'une **connexion initiale** avec le poste de travail ;
  - ◇ il reçoit, sur cette **connexion initiale**, une ligne contenant le TSAP de la machine que le client veut atteindre de l'autre côté du firewall :

    `adresseIP:numéroPort\r\n`
  - ◇ le serveur récupère cette ligne et l'analyse pour établir une **nouvelle connexion** vers le TSAP indiqué ;
  - ◇ à partir de cet instant :
    - ⋆ tout ce que le poste de travail envoie sur la **connexion initiale** est renvoyer sur la **nouvelle connexion** ;
    - ⋆ tout ce que le serveur reçoit sur la **nouvelle connexion** est renvoyé vers le poste de travail sur la **connexion initiale** ;
  - ◇ *vous ne vous occuperez par de la terminaison de ces deux connexions.*

a. indiquez la configuration de base du firewall pour :
   - ◇ permettre le travail du serveur logiciel ;
   - ◇ empêcher les postes de travail de traverser le firewall ;

   On peut bloquer tout le trafic en provenance du réseau à l'aide de la règle suivante :

```
$ sudo iptables -t filter -A FORWARD -s 10.10.0.0/16 -j DROP
```

   Pour autoriser explicitement les connexions vers le routeur en excluant toute autre connexion (attention à la connexion sécurisée nécessaire à l'administration du routeur) :

```
$ sudo iptables -t filter -P INPUT DROP
$ sudo iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -t filter -A INPUT -p tcp --dport 6800 -m state --state NEW -j ACCEPT
```

b. Donnez le programme Python réalisant le travail du serveur logiciel.

```
#!/usr/bin/python
# coding= utf-8
import sys,os,socket,re

def lire_ligne(desc):
    ligne = ''
    while 1:
        car = desc.recv(1)
        if not car : break
        if car == '\n' :
            break
        ligne += car
    return ligne

adresse_hote = ''
numero_port = 6800
ma_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM, socket.IPPROTO_TCP)
ma_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,1)
ma_socket.bind((adresse_hote, numero_port))
ma_socket.listen(socket.SOMAXCONN)
# Expression régulière pour l'analyse du TSAP
re_tsap = re.compile(r'^(\d+\.\d+\.\d+\.\d+):(\d+)')

while 1:
   (nouvelle_connexion, depuis) = ma_socket.accept()
   # Création d'un nouveau processus partageant la connexion
   pid = os.fork()

   if (not pid) :
      # Processus enfant
      ligne = lire_ligne(nouvelle_connexion)
      resultat = re_tsap.search(ligne)
      tsap_relais = resultat.groups()
      socket_relais = socket.socket(socket.AF_INET, socket.SOCK_STREAM, socket.IPPROTO_TCP)
      socket_relais.connect(tsap_relais)
      pid2 = os.fork()
      if pid2 :
         while 1:
            donnees = nouvelle_connexion.recv(1024)
            socket_relais.sendall(donnees)
         nouvelle_connexion.close()
         socket_relais.close()
         sys.exit()
      else :
         while 1:
            donnees = socket_relais.recv(1024)
            nouvelle_connexion.sendall(donnees)
         socket_relais.close()
         nouvelle_connexion.close()
         sys.exit()
ma_socket.close()
```

c. Il existe une ancienne version du logiciel client sur certains postes de travail qui utilise le même protocole **mais** qui utilise le port de destination 7800 pour se connecter au firewall.
Est-il possible, à l'aide du firewall, de les rediriger automatiquement vers la nouvelle version du serveur utilisant le port 6800 ? Si oui, indiquez comment.

À l'aide de la règle REDIRECT :

```
$ sudo iptables -t nat -A PREROUTING -p tcp --dport 7800 -j REDIRECT --to-ports 6800
```